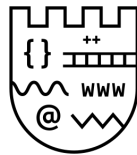


Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Σχολή Θετικών Επιστημών



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εργασία στο μάθημα της Κρυπτογραφίας

Φτιάκας Σωτήριος ΑΕΜ: 3076

Μπάρμπας Γρηγόριος ΑΕΜ: 3108

10 Μαΐου 2020

Περιεχόμενα

Περίληψη	2
Θέμα 1	3
Θέμα 2	3
Θέμα 3	3
Θέμα 4	3
Θέμα 5	3
Θέμα 6	4
Θέμα 7	4
Θέμα 8	7
Θέμα 9	8
Θέμα 10	14
Θέμα 11	14
Θέμα 12	14
Θέμα 13	14
Θέμα 14	16
Θέμα 15	16

Θέμα 16	16
Θέμα 17	16
Θέμα 18	16
Θέμα 19	16
Θέμα 20	16
Θέμα 21	16
Θέμα 22	16
Θέμα 23	17
Θέμα 24	17
Θέμα 25	17
Θέμα 26	18
Θέμα 27	18
Θέμα 28	18
Θέμα 29	18
Θέμα 30	18
Θέμα 31	18

Περίληψη

.....

Θέμα 1

Θέμα 2

Θέμα 3

Θέμα 4

Θέμα 5

Dictionary Attack

```
1 # Import of library
2 from zipfile import ZipFile
3
4 # Variable names
5 zip_file = 'test_zip.zip'
6 eng_dict = 'english.txt'
7 check = 0
8
9 # Move the words from txt file to a list
10 with open(eng_dict) as ed:
11     content = ed.readlines()
12 # you may also want to remove whitespace characters like '\n' at the end of
    each line
13 content = [x.strip() for x in content]
14
15 # Brute Force until we find the right password
16 i = 1
17 print('Hacking', end='')
18 for password in content:
19     if (i%5000==1):
```

```

20     print('.', end='', flush=True)
21     try:
22         with ZipFile(zip_file) as zf:
23             zf.extractall(pwd=bytes(password, 'utf-8'))
24             check = 1
25     except Exception:
26         check = 0
27         pass
28     if check == 1:
29         break;
30     i = i + 1
31
32 print('')
33 print('FILE UNLOCKED')

```

Θέμα 6

Θέμα 7

Shift Operator with XOR

m: 16-bits

$$c = m \oplus (m \ll 6) \oplus (m \ll 10)$$

Όπου $m \ll a$ είναι κύλιση προς τα αριστερά κατά a -bits.

Για μήνυμα m και κλειδί k ισχύει: Αν $c = m \oplus k$, τότε $m = c \oplus k$

Επιπλέον, στην αρχική μας συνάρτηση κρυπτογράφησης, μπορούμε να κυλίσουμε και τα δύο μέλη ταυτόχρονα.

$$(c \ll 2) = (m \oplus (m \ll 6) \oplus (m \ll 10)) \ll 2$$

$$\Leftrightarrow (c \ll 2) = (m \ll 2) \oplus (m \ll 8) \oplus (m \ll 12)$$

Σημείωση: το $x \ll i$ θα συμβολίζεται ως x_i για ευκολία.

Συνεπώς θα έχουμε:

$$c_0 = m_0 \oplus m_6 \oplus m_{10} \quad (1)$$

$$c_2 = m_2 + m_8 + m_{12} \Rightarrow m_8 = m_2 + m_{12} + c_2 \quad (4)$$

$$c_4 = m_4 + m_{10} + m_{14} \Rightarrow m_{10} = m_4 + m_{14} + c_4 \quad (2)$$

$$c_6 = m_6 + m_{12} + m_0 \quad (5)$$

$$c_8 = m_8 + m_{14} + m_2$$

$$c_{10} = m_{10} + m_0 + m_4$$

$$c_{12} = m_{12} + m_2 + m_6$$

$$c_{14} = m_{14} + m_4 + m_8 \Rightarrow m_{14} + m_4 = m_8 + c_{14} \quad (3)$$

Ξεκινώντας από την (1) έχουμε διαδοχικά:

$$c_0 = m_0 \oplus m_6 \oplus m_{10}$$

$$(2) \Rightarrow c_0 = m_0 + m_6 + m_4 + m_{14} + c_4$$

$$(3) \Rightarrow c_0 + c_4 = m_0 + m_6 + m_8 + c_{14}$$

$$(4) \Rightarrow c_0 + c_4 + c_{14} = m_0 + m_6 + m_2 + m_{12} + c_2$$

$$(5) \Rightarrow c_0 + c_4 + c_{14} + c_2 = m_2 + c_6$$

$$\Rightarrow c_0 + c_4 + c_{14} + c_2 + c_6 = m_2 \quad (6)$$

Κάνουμε κύλιση και στα δύο μέρη του (6) προς τα δεξιά και έχουμε:

$$m_0 = c_{14} + c_2 + c_{12} + c_0 + c_4$$

και άρα τελικά έχουμε:

$$m_0 = c_0 + c_2 + c_4 + c_{12} + c_{14}$$

Κώδικας σε python

```
1 import random
2
3 # Create a random message of "n" bits.
4 def create_m(n):
5     message = []
6     for i in range(n):
7         bit = bool(random.getrandbits(1))
8         if bit == True:
9             message.append(1)
10        else:
11            message.append(0)
12
13    return message #returns a list of bits
14
15 # Left rotation of a message by n.
16 def rotate(message, n):
17     return message[n:] + message [:n] #returns a list of bits
18
19 # XOR function between a message and a key.
20 def xor(message, key):
21     c = [0 for i in range(len(message))]
22     for i in range(len(message)):
23         temp = 0
```

```

24         if (message[i] != key[i]):
25             c[i] = 1
26
27         return c #returns a list of bits
28
29 # Message Encryption
30 m0 = create_m(16) # original message
31 m6 = rotate(m0, 6)
32 m10 = rotate(m0, 10)
33
34 c0 = xor(m0, xor(m6,m10)) # c = m (+) (m<<6) (+) (m<<10)
35
36 print("Original Message: ")
37 print(m0)
38
39 print("Encrypted Message: ")
40 print(c0)
41
42 # Message Decryption
43 c2 = rotate(c0, 2)
44 c4 = rotate(c0, 4)
45 c12 = rotate(c0, 12)
46 c14 = rotate(c0, 14)
47 decr = xor(c0, xor(c2, xor(c4, xor(c12,c14)))) # d = c (+) (c<<2) (+) (c<<4)
         (+) (c<<12) (+) (c<<14)
48
49
50 print("Decrypted Message: ")
51 print(decr)
52
53 print("Original Message: ")
54 print(m0)

```

Θέμα 8

Θέμα 9

Entropy

Y/X	0	1	2
0	$1/7$	$1/7$	$1/7$
1	0	$1/7$	$1/7$
2	$2/7$	0	0

Αρχικά υπολογίζουμε

$$p_x(X=0) = \sum_y p_{X,Y}(0,y) = \frac{3}{7}$$

$$p_x(X=1) = \sum_y p_{X,Y}(1,y) = \frac{2}{7}$$

$$p_x(X=2) = \sum_y p_{X,Y}(2,y) = \frac{2}{7}$$

$$p_y(Y=0) = \sum_x p_{X,Y}(x,0) = \frac{3}{7}$$

$$p_y(Y=1) = \sum_x p_{X,Y}(x,1) = \frac{2}{7}$$

$$p_y(Y=2) = \sum_x p_{X,Y}(x,2) = \frac{2}{7}$$

Ισχύει ότι:

$$H(X) = - \sum_x p_X(x) \log_2 p_X(x)$$

Επομένως

$$\begin{aligned}H(X) &= -\frac{3}{7} \log_2 \frac{3}{7} - \frac{2}{7} \log_2 \frac{2}{7} - \frac{2}{7} \log_2 \frac{2}{7} \\&= -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{2}{7} \\&\simeq 1.5566567074628228\end{aligned}$$

$$\begin{aligned}r(Q) &= -\frac{3}{7} \log_2 \frac{3}{7} - \frac{2}{7} \log_2 \frac{2}{7} - \frac{2}{7} \log_2 \frac{2}{7} \\&= -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{2}{7} \\&\simeq 1.5566567074628228\end{aligned}$$

Επίσης έχουμε τον εξής τύπο

$$H(X, Y) = - \sum_x \sum_y p(x, y) \log_2 p(x, y)$$

Άρα θα έχουμε:

$$\begin{aligned}H(X, Y) &= -p(0, 0) \log_2 p(0, 0) - p(0, 1) \log_2 p(0, 1) - p(0, 2) \log_2 p(0, 2) - \\&\quad p(1, 0) \log_2 p(1, 0) - p(1, 1) \log_2 p(1, 1) - p(1, 2) \log_2 p(1, 2) - \\&\quad p(2, 0) \log_2 p(2, 0) - p(2, 1) \log_2 p(2, 1) - p(2, 2) \log_2 p(2, 2) \\&\simeq 2.5216406363433186\end{aligned}$$

Θα υπολογίσουμε την $H(Y|X)$. Χρειαζόμαστε αρχικά τα παρακάτω,

$$p_{Y|X}(y = 0|x = 0) = \frac{p_{X,Y}(0, 0)}{p_X(0)} = \frac{\frac{1}{7}}{\frac{3}{7}} = \frac{1}{3}$$

$$p_{Y|X}(y = 1|x = 0) = \frac{p_{X,Y}(0, 1)}{p_X(0)} = \frac{0}{\frac{3}{7}} = 0$$

$$p_{Y|X}(y = 2|x = 0) = \frac{p_{X,Y}(0, 2)}{p_X(0)} = \frac{\frac{2}{7}}{\frac{3}{7}} = \frac{2}{3}$$

$$p_{Y|X}(y = 0|x = 1) = \frac{p_{X,Y}(1, 0)}{p_X(1)} = \frac{\frac{1}{7}}{\frac{2}{7}} = \frac{1}{2}$$

$$p_{Y|X}(y = 1|x = 1) = \frac{p_{X,Y}(1, 1)}{p_X(1)} = \frac{\frac{1}{7}}{\frac{2}{7}} = \frac{1}{2}$$

$$p_{Y|X}(y = 2|x = 1) = \frac{p_{X,Y}(1, 2)}{p_X(1)} = \frac{0}{\frac{2}{7}} = 0$$

$$p_{Y|X}(y = 0|x = 2) = \frac{p_{X,Y}(2, 0)}{p_X(2)} = \frac{\frac{1}{7}}{\frac{2}{7}} = \frac{1}{2}$$

$$p_{Y|X}(y = 1|x = 2) = \frac{p_{X,Y}(2, 1)}{p_X(2)} = \frac{\frac{1}{7}}{\frac{2}{7}} = \frac{1}{2}$$

$$p_{Y|X}(y = 2|x = 2) = \frac{p_{X,Y}(2, 2)}{p_X(2)} = \frac{0}{\frac{2}{7}} = 0$$

Τώρα πρέπει να υπολογίσουμε τα παρακάτω:

$$\begin{aligned}H(Y|X=0) &= - \sum_y p_{Y|X}(y|x=0) \log_2 p_{Y|X}(y|x=0) \\&= -(\frac{1}{3} \log_2 \frac{1}{3} + 0 + \frac{2}{3} \log_2 \frac{2}{3}) \\&= -(\frac{1}{3} \log_2 \frac{1}{3} + \frac{2}{3} \log_2 \frac{2}{3}) \\H(Y|X=1) &= - \sum_y p_{Y|X}(y|x=1) \log_2 p_{Y|X}(y|x=1) \\&= -(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} + 0) \\&= -\log_2 2 = 1 \\H(Y|X=2) &= - \sum_y p_{Y|X}(y|x=2) \log_2 p_{Y|X}(y|x=2) \\&= -\log_2 2 = 1\end{aligned}$$

Τότε θα έχουμε

$$\begin{aligned}H(Y|X) &= \sum_x p_X(x) H(Y|X=x) \\&= p_X(0)H(Y|X=0) + p_X(1)H(Y|X=1) + p_X(2)H(Y|X=2) \\&\simeq 0.9649839288804954\end{aligned}$$

Γνωρίζουμε επίσης ότι από το θεώρημα της αμοιβαίας πληροφορίας έχουμε

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Άρα έχουμε

$$H(X|Y) = -(H(Y) - H(Y|X) - H(X))$$
$$\simeq 0.9649839288804954$$

Τέλος,

$$\rho = 1 - \frac{H(Y|X)}{H(X)}$$
$$\simeq 0.5916727785823274$$

Κώδικας σε python

```
1 import numpy as np
2 import math
3
4 def create_table(elements):
5     return np.reshape(elements, (int(math.sqrt(len(elements))),int(math.sqrt(
6         len(elements)))))
7
8 def entropy(table):
9     # Calculate pX and pY
10    pX = [sum(i) for i in zip(*table)] #add columns
11    pY = [sum(i) for i in table] #add rows
12
13    # Calculate HX and HY
14    HX = -sum([x*math.log(x,2) for x in pX])
15    HY = -sum([x*math.log(x,2) for x in pY])
16
17    return HX, HY, pX, pY
18
19 def mutual_entropy(table):
20     temp = np.reshape(table, (len(table)**2)) # flatten table (easier to
```

```

iterate)
20     temp = [i for i in temp if i != 0] # remove 0 from list (log(0) is
    undefined)
21
22     HX_Y = -sum([x*math.log(x,2) for x in temp])
23
24     return HX_Y
25
26 def committed_entropy(table):
27     HX, HY, pX, pY = entropy(table)
28
29     temp= []
30     for i in range(len(table)):
31         for j in range(len(table)):
32             temp.append(table[j][i] / pX[i])
33
34     temp = [i if i != 0 else 1 for i in temp] # replace 0 from list (log(0) is
    undefined)
35
    #log(1) = 0 so it will not
    affect our results
36
37     pY_X = create_table(temp)
38
39     HY_X_table = []
40     for row in pY_X:
41         HY_X_table.append(-sum([x*math.log(x,2) for x in row]))
42
43     HY_X = sum([pX[i] * HY_X_table[i] for i in range(len(pX))])
44
45     # Mutual Information:  $I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ 
46     HX_Y = -(HY - HY_X - HX)
47
48     return HX_Y, HY_X

```

```
49
50 def mutual_information(table):
51     HX, _, _, _ = entropy(table)
52     HX_Y, _ = committed_entropy(table)
53
54     IX_Y = HX - HX_Y
55
56     return IX_Y
57
58 elements = [1/7,1/7,1/7,0,1/7,1/7,2/7,0,0]
59 table = create_table(elements)
60
61
62 HX, HY, _, _ = entropy(table)
63 print("Entropy: ")
64 print(HX, HY)
65 print()
66 print("Mutual Entropy: ")
67 print(mutual_entropy(table))
68 print()
69 print("Committed Entropy: ")
70 print(committed_entropy(table))
71 print()
72 print("Mutual Information: ")
73 print(mutual_information(table))
```

Θέμα 10

Θέμα 11

Θέμα 12

Θέμα 13

Chinese Theorem

Έχουμε το σύστημα των γραμμικών ισοδυναμιών

$$x \equiv 9 \pmod{19}$$

$$x \equiv 9 \pmod{12}$$

$$x \equiv 13 \pmod{17}$$

Έχουμε ότι ισχύει $\gcd(12, 17, 19) = 1$, άρα δεν απαιτείται κάποια απλοποίηση.

Για την επίλυση του συστήματος χρησιμοποιούμε το Κινέζικο Θεώρημα Υπολοίπων

Έτσι έχουμε: $m = 17 * 12 * 19 = 3876$

$$M_1 = 228y_1 \equiv 1 \pmod{17} \implies 7y_1 \equiv 1 \pmod{17} \implies y_1 = 5$$

$$M_2 = 323y_2 \equiv 1 \pmod{12} \implies 11y_2 \equiv 1 \pmod{12} \implies y_2 = 11$$

$$M_3 = 204y_3 \equiv 1 \pmod{19} \implies 14y_3 \equiv 1 \pmod{19} \implies y_3 = 15$$

Τώρα πολλαπλασιάζουμε και προσθέτουμε:

$$\begin{aligned} x &= 9 * 228 * 5 + 9 * 323 * 11 + 13 * 204 * 15 \\ &= 82017(1) \end{aligned}$$

Παρατηρούμε ότι η (1) γράφεται,

$$x = 82017 = 621 + 3876k, k \in \mathbb{Z}$$

Για $k = 0$ έχουμε λύση το $x = 621$

Κώδικας σε python

```
1 # Brute force algorithm to test results
2 for x in range(1,1000):
3     if ((x % 17) == 9) and ((x % 12) == 9) and ((x % 19) == 13):
4         print(x)
```

Θέμα 14

Θέμα 15

Θέμα 16

Θέμα 17

Θέμα 18

9e94b15ed312fa42232fd87a55db0d39

Θέμα 19

Θέμα 20

Θέμα 21

Θέμα 22

Αρχικά θέλουμε να αποδείξουμε ότι οι αριθμοί της μορφής $4n+3$ δεν είναι τέλεια τετράγωνα. Αρχικά για $n \leq 0$ εύκολα παρατηρούμε ότι ισχύει η παραπάνω πρόταση.

Αυτό συμβαίνει καθώς για $n = 0$ έχουμε το 3 το οποίο δεν είναι τέλειο τετράγωνο και για $n < 0$ το $4n + 3$ είναι αρνητικός.

Έστω ότι,

$$4n + 3 = a^2, \quad n, a \in \mathbb{N}^*(1)$$

Εφόσον $a \in \mathbb{N}^*$, τότε μπορούμε να πούμε ότι $a = 2k + 1$, $k \in \mathbb{N}$.

Αντικαθιστώντας το a στην (1) έχουμε:

$$\begin{aligned} 4n + 3 &= (2k + 1)^2 \implies 4n + 3 = 4k^2 + 4k + 1 \\ &\equiv 4k^2 + 4k - 4n = 2 \\ &\equiv 2(k^2 + k - n) = 1 \\ &\equiv k^2 + k - n = \frac{1}{2} \end{aligned}$$

Το οποίο είναι άτοπο καθώς $k, n \in \mathbb{N}^*$ και άρα το $k^2 \in \mathbb{N}^*$ αλλά και όλη η παράσταση $k^2 + k - n \in \mathbb{N}$, εφόσον είναι άθροισμα των φυσικών αριθμών k^2, k και $-n$.

Συνεπώς και η αρχική ισοδύναμη υπόθεση είναι άτοπη, οπότε το $4n + 3$ δεν είναι τέλειο τετράγωνο.

Για το δεύτερο ερώτημα παρατηρούμε ότι όλοι αριθμοί

$$11, 111, \dots, 111 \cdots 111, \dots$$

μπορούν να γραφούν στην μορφή $(4n + 3) + 10^q$, συνεπώς αυτό το σύνολο αριθμών δεν θα έχει τέλειο τετράγωνο.

Θέμα 23

Θέμα 24

Θέμα 25

Θέμα 26

Θέμα 27

Θέμα 28

Θέμα 29

Θέμα 30

Υπόθεση:

$$N > 2$$

$$N = p_1 p_2 \dots p_k$$

$$p_j - 1 \mid N - 1 \quad \forall j$$

Απόδειξη:

$$\text{Έστω } \gcd(a, N) = 1.$$

Από το θεώρημα του Fermat, $\forall j$, έχουμε $a^{p_j} \equiv 1 \pmod{p_j}$.

Εφόσον $p_j - 1 \mid N - 1$, και άρα $a^{N-1} \equiv 1 \pmod{p_j}$.

Δηλαδή το $a^{N-1} - 1$ είναι πολλαπλάσιο κάθε p_j .

Συνεπώς $a^{N-1} \equiv 1 \pmod{N}$.

Θέμα 31

Αναφορές